Applied Deep Learning COMS4995 Custom Course Project | FS22

Deep Learning for Finance:
Using stock market news and daily close values
to predict next-day increase or decrease

Erin Josephine Donnelly
ejd2170

Introduction

Machine learning for finance has some tried-and-true methods

- Time series data with seasonality → RNNs and LSTM
- Tabular data → supervised learning: regression and tree-based models
- News articles → semantic analysis

The challenge: **combining the multiple mediums of finance-related data into a multi-stage model**

- Two inputs: **news** and **stock** data
- Process with individual LSTM layers, concatenate output, and continue with Dense layers
- **Binary classification:** "*Will the stock's close value increase or decrease the following day?*"

Objectives

1) **Minimal goa**l: Explore DNNs, RNNs, and LSTM approaches to predict decisions for one stock

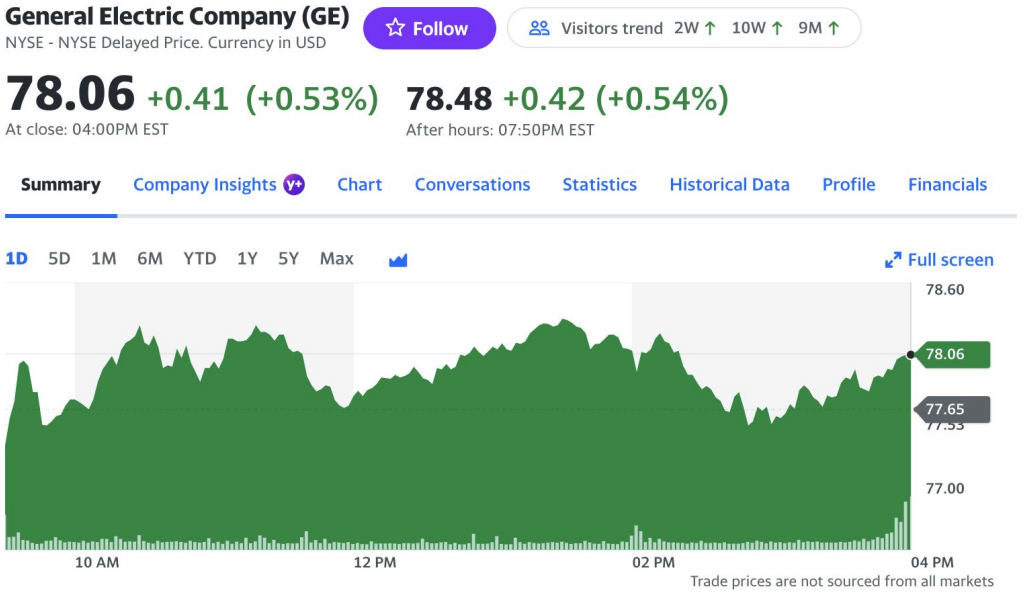
2) **Expected goal**: Extend the minimal goal by including text input from news sources, and develop a model to predict decisions for multiple stocks

3) **Stretch goal:** Extend the expected goal by implementing scraping to obtain more diverse and more current text sources for stronger sentiment analysis

I am pleased to say that I was able to fulfill the third version **stretch goal** of the project, including two methods of scraping to get the most complete news data from Yahoo Finance.

Project phases

1) EDAV
2) Data acquisition, scraping, processing
3) Preliminary models and exploration
4) Final multi-stage model
5) Demo

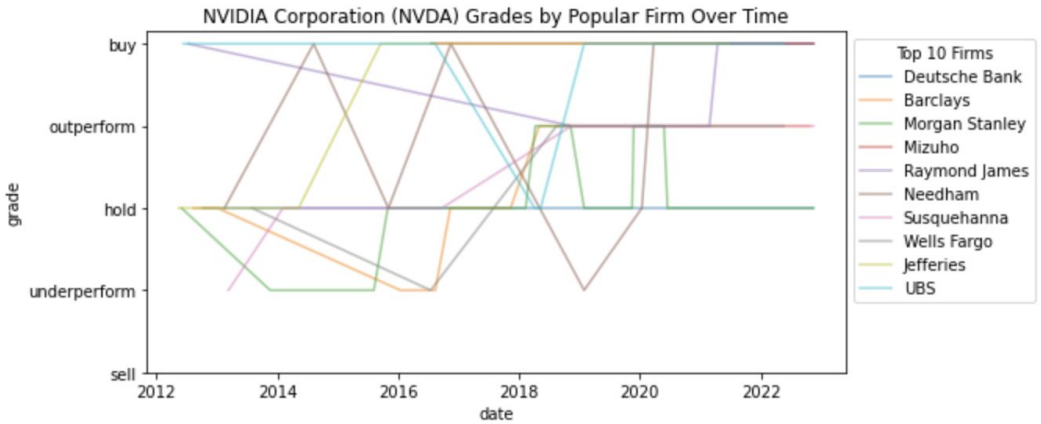


Yahoo Finance Stock Market News: https://finance.yahoo.com/quote/GE?p=GE

EDAV

Elements explored and visualized:

- Grades over time , by stock and company
- Stock trends over time
- Value comparison: close, open, difference
- News availability through Yahoo FInance API



This step also involved domain research, understanding stock vocabulary, etc.

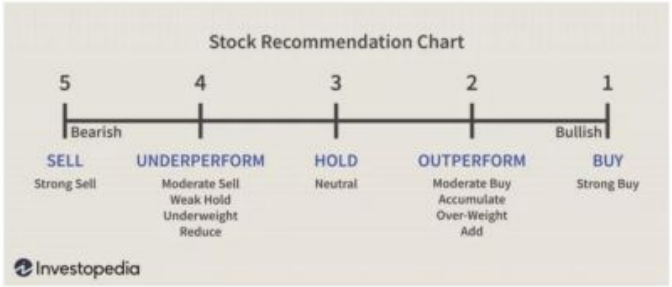


Image by Julie Bang © Investopedia 2020

The Scale of Ratings: https://www.investopedia.com/financial-edge/0512/understanding-analyst-ratings.aspx

Data and scraping

Two inputs:

- **News data**: scraped from sources obtained through the Yahoo Finance API and Yahoo Finance Stock Market News search HTML code
- **Stock data**: downloaded through the Yahoo Finance API

Labels:

- Problem reduced to binary classification
- Next-day decrease: 0
- Next-day increase (or maintenance): 1

Scraping:

- Used BeautifulSoup
- **5487 articles** across **161 stocks**, primarily from **September 2022 – December 2022**

```
print(f'Number of urls found: {np.sum([len(all_urls[idx]) for idx in all_urls.keys()])}')
for index in all_urls.keys():
    print(index)
    for link in all_urls[index][:2]:
        print('\t', link)
```

```
Number of urls found: 2491
RS
	 http://www.finance.yahoo.com/news/3-reasons-why-reliance-steel-174505323.html
	 http://www.finance.yahoo.com/news/reliance-steel-rs-forms-hammer-145502989.html
W
	 http://www.finance.yahoo.com/m/80039d6d-86bb-38e5-bb3f-c453bae126e1/where-will-wayfair-stock-be.html
	 http://www.finance.yahoo.com/m/df19f68b-57b0-3f82-9be3-c8a355b8454a/wayfair-gets-a-new-bull-it.html
TAL
	 http://www.finance.yahoo.com/news/tal-education-group-tal-stock-144002474.html
	 http://www.finance.yahoo.com/news/4-promising-chinese-stocks-buy-130101039.html
X
	 http://www.finance.yahoo.com/m/1e370eb5-7ca6-3ae0-91cc-6f3d05384e94/stocks-extend-slump-twitter-.html
	 http://www.finance.yahoo.com/video/stocks-moving-hours-adobe-united-220540829.html
GOTU
	 http://www.finance.yahoo.com/news/gaotu-techedu-announces-third-quarter-060000154.html
	 http://www.finance.yahoo.com/news/gaotu-techedu-announces-receipt-nyse-123000091.html
DAL
	 http://www.finance.yahoo.com/m/062b8b5e-81cf-37e4-91d3-35509519a1b7/barron%E2%80%99s-10-favorite-stocks.html
	 http://www.finance.yahoo.com/news/delta-air-lines-dal-stock-230011253.html
```

Data and Scraping

An important note on data quality and availability:

While there is ample stock data, there is **insufficient news data to train a deep learning model with excellent predictive results on the test set**.

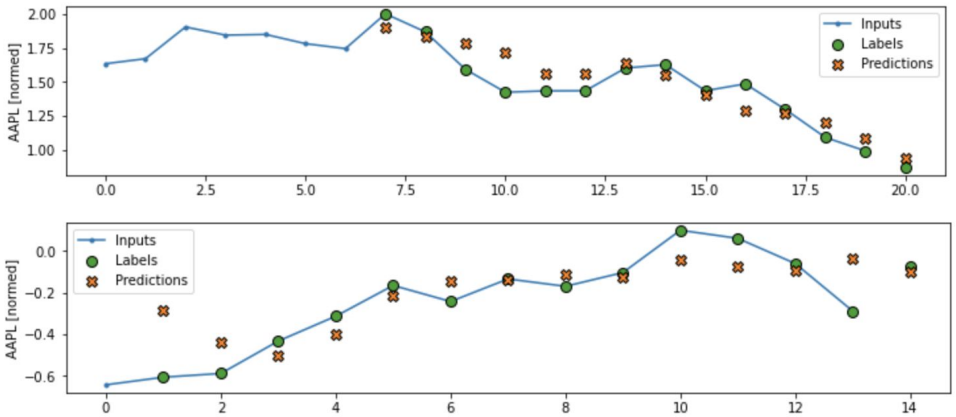
- This course has set a precedent for us to **do good work with the data we have** while **considering improvements to be made in future work**
- With more news data over longer periods of time, we can **extend the labels to reflect more stable stock changes**, such as taking the average value over the following week or month instead of a single day
- Limitations are discussed throughout the final model notebook

After all, the goal of this project is not to develop a new state-of-the-art stock predictor, but to **develop an innovative approach that answers an interesting question and looks beyond off-the-shelf solutions**

Preliminary models

Time series forecasting

- Considered **stock data only**
- Modified from TensorFlow tutorial: https://www.tensorflow.org/tutorials/structured_data/time_series
- Showed promise, but depended on seasonality



Single-phase models:

- Used desired multi-phase models with tuned hyperparameters, but restricted input to only one source
- **News only**: like semantic analysis with next-day stock change acting as *"free labels"*; overfits severely to training data
- **Stocks only**: efficient models struggle to learn training data, but do not overfit

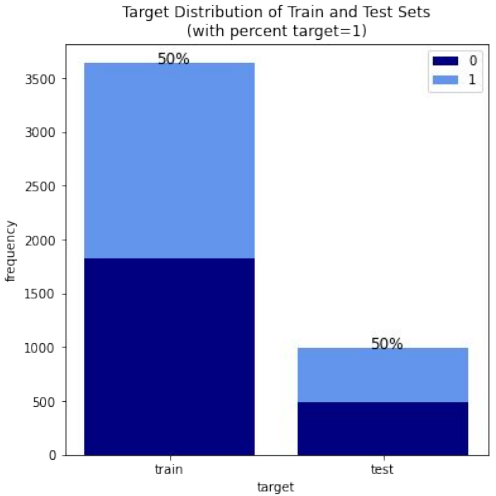
The multi-phase model will act as a compromise: learning generalizable patterns in the training data without extreme overfitting. Single-phase models inspired more focus on stock data over news data.

Multi-stage model

Several **hyperparameters** to decide:

- History of data considered
- Train-test-split
- Length and timesteps of stock sequences
- Commonality between stocks and dates
- Uniqueness of examples per date
- Balancing the dataset
- Text processing and vectorization
- Model inputs
- Model architecture
- Embedding and layer dimensionality
- Dropout
- Learning rate
- Epochs

```
N_DAYS_IN_TEST = 7
N_STOCK_UNITS = 2
STOCK_UNIT = 'mo'
N_SEQ_WEEKS = 4
N_TIMESTEPS = 3

UNIQUE_URL = True
ONE_PER_DAY = False
MIN_STOCK_FREQ = None

REMOVE_STOPWORDS = True
BALANCE_DATASET = True

USE_NEWS_MODEL = True
USE_STOCKS_MODEL = True

NEWS_EMBEDDING_DIM = 128
NEWS_USE_BIDIRECTIONAL = True
NEWS_UNITS = [16]
NEWS_DROPOUT = 0.2

STOCK_UNITS = [32, 64]
STOCK_DROPOUT = 0.1

DENSE_UNITS = [32, 32, 1]

LR = 1e-4

BUFFER_SIZE = 1000
BATCH_SIZE = 32

N_EPOCHS = 15

# hyperparameters chosen using algorithms later:
# MAX_SEQ_LEN
# VOCAB_SIZE
```



Multi-stage model

News model:

- Vectorized input
  - input_dim = 10000
  - input_length = 490
- Embedding layer
  - input_dim = 128
- Bidirectional LSTM layer
  - units = 16
  - return_sequences = False
  - dropout = 0.2

Stock model:

- 3-dimensional timestep input
  - 3 values per subsequence
- LSTM layer
  - units = 32
  - return_sequences = True
  - dropout = 0.1
- LSTM layer
  - units = 64
  - return_sequences = False
  - dropout = 0.1

Merged:

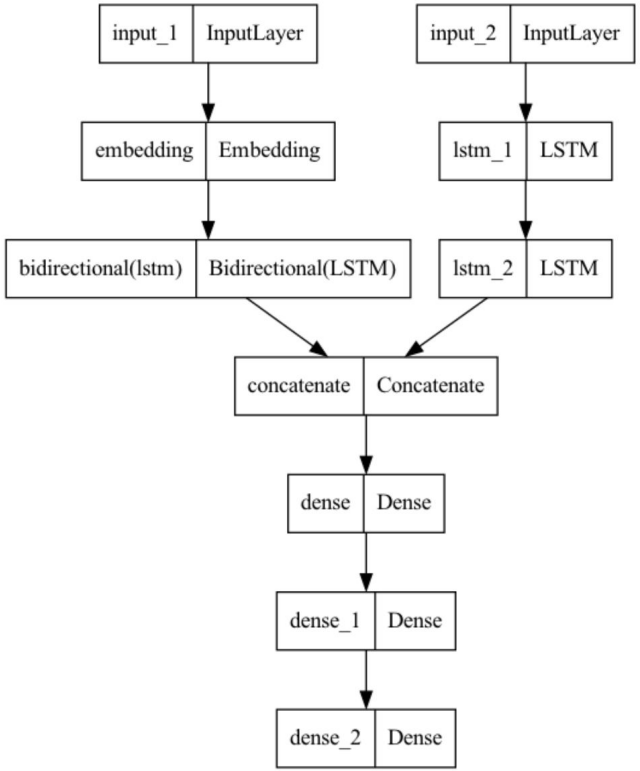
- Concatenated input
- Dense layer
  - units = 32
  - activation = ‘relu’
- Dense layer
  - units = 32
  - activation = ‘relu’
- Dense layer
  - units = 1
  - activation = ‘sigmoid’

Multi-stage model

Training time: **~18s per epoch** (15 epochs)

```
model = build_and_compile_model()
model.summary()
```

```
Model: "model"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_1 (InputLayer)           [(None, 490)]        0           []

 input_2 (InputLayer)           [(None, 17, 3)]      0           []

 embedding (Embedding)          (None, 490, 128)     1280000     ['input_1[0][0]']

 lstm_1 (LSTM)                  (None, 17, 32)       4608        ['input_2[0][0]']

 bidirectional (Bidirectional)  (None, 32)           18560       ['embedding[0][0]']

 lstm_2 (LSTM)                  (None, 64)           24832       ['lstm_1[0][0]']

 concatenate (Concatenate)      (None, 96)           0           ['bidirectional[0][0]',
                                                                  'lstm_2[0][0]']

 dense (Dense)                  (None, 32)           3104        ['concatenate[0][0]']

 dense_1 (Dense)                (None, 32)           1056        ['dense[0][0]']

 dense_2 (Dense)                (None, 1)            33          ['dense_1[0][0]']

==================================================================================================
Total params: 1,332,193
Trainable params: 1,332,193
Non-trainable params: 0
__________________________________________________________________________________________________
```



Multi-stage model

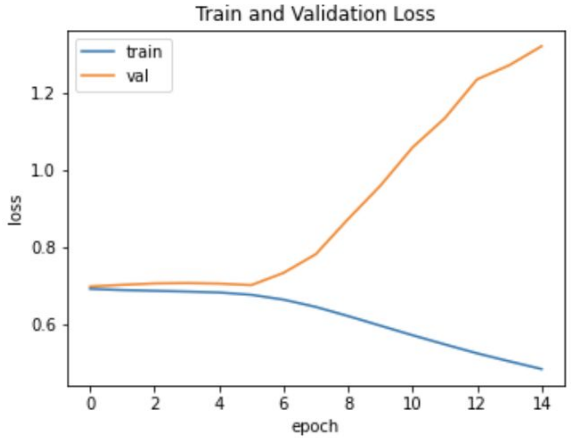
Standard metrics: **train and test loss and accuracy**

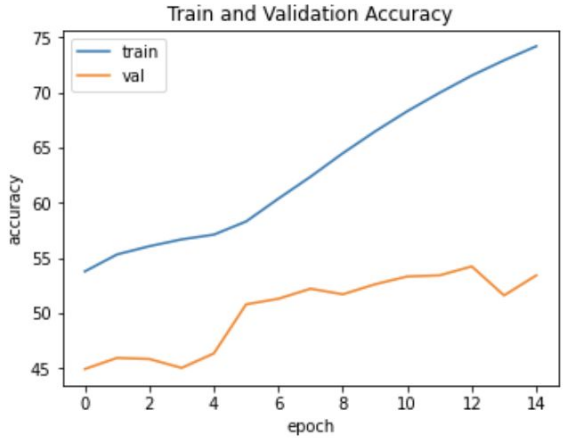
- BinaryCrossentropy, BinaryAccuracy
- Train: Loss = 0.48, Accuracy = 74.18%
- Test: **Loss = 1.32, Accuracy = 53.43%**

Additional metric:

average score grouped by stock and date

- Test accuracy with rounding threshold = 0.5: **51.89%**





Demo

Preparation

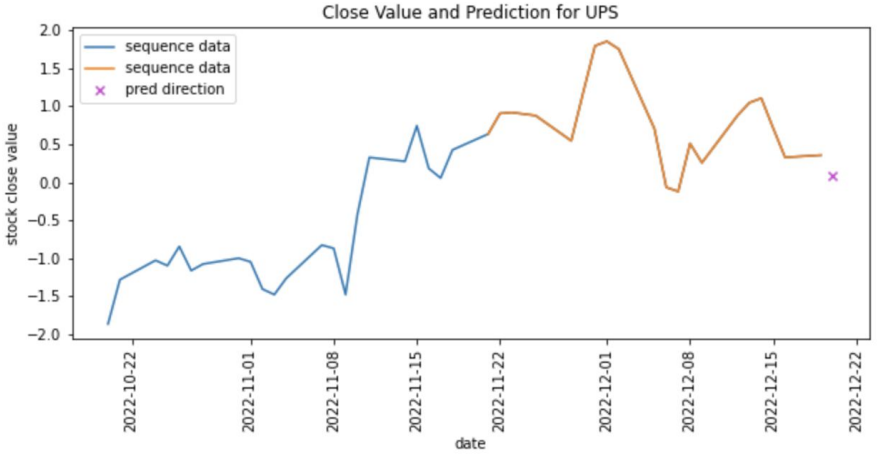
- Loads and merges news and stock data
- Performs processing, train-test-split, dataset balancing, etc. determines model architecture **using the same hyperparameters and functions as during training**
- Restores saved model weights

Prompting the user:

- Supply a stock
- Supply a prediction date
- Obtain predictions
- Visualize the predicted and true results
  or
  infer future stock behavior

```
predict_and_display(prediction_date, stock_demo_df)
```

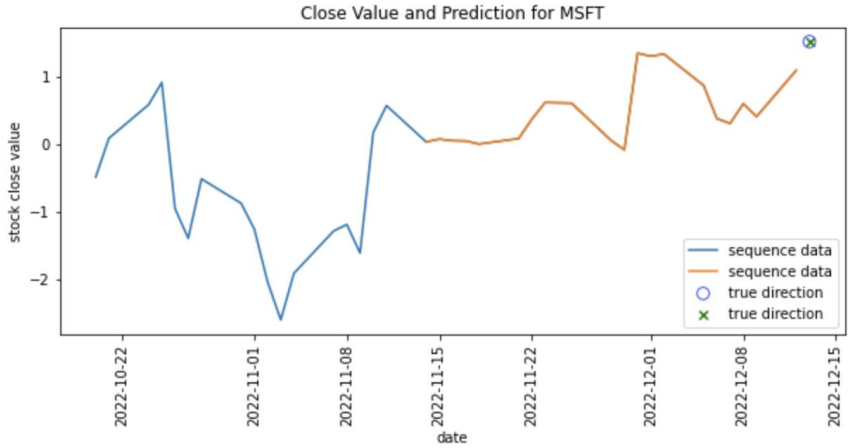
```
Prediction:
        0.00 --> 0
        Stock UPS will **decrease** from 2022-12-19 to 2022-12-20

Check the stock's close value on 2022-12-20 to see if the prediction was right!
```



Demo

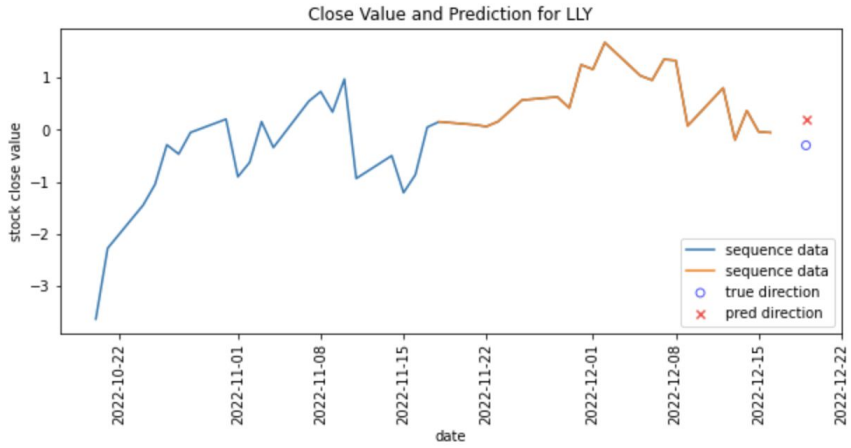
```
predict_and_display(prediction_date, stock_demo_df)
```

```
Prediction:
        0.91 --> 1
        Stock MSFT will **increase** from 2022-12-12 to 2022-12-13

True outcome:
        1
        Stock MSFT **increased** from 2022-12-12 to 2022-12-13

The predictions was correct!
```




```
predict_and_display(prediction_date, stock_demo_df)
```

```
Prediction:
        0.89 --> 1
        Stock LLY will **increase** from 2022-12-16 to 2022-12-19

True outcome:
        0
        Stock LLY **decreased** from 2022-12-16 to 2022-12-19

The prediction was wrong this time.
```



Conclusion

Results:

	Loss	Accuracy	Average Grouped Accuracy
Train	0.48	74.18%	—
Test	1.32	53.43%	51.89%

Observations:

- Overfitting to training data, though improved from news-only models
- **Shows promise**, but this model acts as a **proof of concept**; much more work can be done to **improve results**

Improvements and future work:

- Acquire more news data over a longer period of time
- Contribute more data to labels (ex. Increasing or decreasing on average over the following week or month
- Refine selection of news articles for more consistent labels, confirmed relevance of stocks and articles, etc.
- Obtain more sophisticated news labeling such as domain-specific semantic analysis
- Use news data sources beyond Yahoo Finance, like the Financial Times or Harvard Business Review
- Continue model selection and hyperparameter tuning
- Redefine the problem as regression to for stock value prediction beyond simple increase or decrease

Thank you

Deep Learning for Finance:
Using stock market news and daily close values
to predict next-day increase or decrease

Erin Josephine Donnelly
ejd2170

GitHub repository:
https://github.com/ejosied/dl-for-finance